Hybrid integration of rules and ontologies:

A constraint-based framework

Jakob Henriksson, Jan Małuszyński

RuleML 2006, Athens, GA
The objective

• Define a scheme that
  **from given**
  – Rule language \( R \) (e.g. Datalog, Xcerpt)
  – Logical language \( S \) (e.g. OWL-DL, …)

**constructs**
– A language \( R_S \) integrating \( R \) and \( S \):
  • Syntax, Semantics of \( R_S \): from syntax and semantics of \( R \) and \( S \)
  • A (complete) reasoner for \( R_S \)
    by interfacing *existing* reasoners of \( R \) and \( S \)
Outline

• Motivating example
• The scheme
  – Principles and restrictions
• Reusing reasoners
  – Datalog + OWL-DL
  – Xcerpt + OWL-DL
• Practical reasoning
  – Eager interaction
• Conclusions
• Future work
Motivating example

Rule component $\Pi$:

$r_1$: \text{price-in-usa(X,high) \ :- \ made-by(X,Y), \ NoFellowCompany(Y).}$

$r_2$: \text{price-in-usa(X,high) \ :- \ made-by(X,Y), \ AmericanAssociate(Y), \ monopoly-in-usa(Y,X).}$

$r_3$: \text{made-by(a,b).}$

$r_4$: \text{monopoly-in-usa(b,a).}$

DL component $\Sigma$:

$T-Box$:

- $\text{European \cap American} \subseteq \bot$
- $\text{NoFellowCompany} \equiv \forall \text{associate}. \neg \text{American}$
- $\text{EuropeanAssociate} \equiv \exists \text{associate}. \text{American}$
- $\text{AmericanAssociate} \equiv \exists \text{associate}. \text{American}$
- $\text{InternationalCompany} \equiv \text{EuropeanAssociate} \cup \text{AmericanAssociate}$

$A-Box$:

InternationalCompany(b)

Motivating example

Rule component $\Pi$:

- $r_1$: `price-in-usa(X, high) :- made-by(X, Y), NoFellowCompany(Y).`
- $r_2$: `price-in-usa(X, high) :- made-by(X, Y), AmericanAssociate(Y), monopoly-in-usa(Y, X).`
- $r_3$: `made-by(a, b).`
- $r_4$: `monopoly-in-usa(b, a).`

DL component $\Sigma$:

$T$-Box:

- European $\cap$ American $\subseteq \bot$
- NoFellowCompany $\equiv \forall \text{associate.} \neg\text{American}$
- EuropeanAssociate $\equiv \exists \text{associate.}\text{American}$
- AmericanAssociate $\equiv \exists \text{associate.}\text{American}$
- InternationalCompany $\equiv \text{EuropeanAssociate} \cup \text{AmericanAssociate}$

$A$-Box:

- InternationalCompany(b)

Constraining extent of head predicates with ...

... constraint domain.
Motivating example

Rule component $\Pi$:

$r_1$: price-in-usa(X, high) :- made-by(X, Y), NoFellowCompany(Y).

$r_2$: price-in-usa(X, high) :- made-by(X, Y), AmericanAssociate(Y), monopoly-in-usa(Y, X).

$r_3$: made-by(a, b).

$r_4$: monopoly-in-usa(b, a).

DL component $\Sigma$:

$T$-Box:

European $\cap$ American $\subseteq \bot$

NoFellowCompany $\equiv \forall \text{associate}. \neg$American

EuropeanAssociate $\equiv \exists \text{associate}. \text{American}$

AmericanAssociate $\equiv \exists \text{associate}. \text{American}$

InternationalCompany $\equiv \text{EuropeanAssociate} \cup \text{AmericanAssociate}$

$A$-Box:

InternationalCompany(b)

$\Pi \cup \Sigma \models \text{price-in-usa}(a, \text{high})$?
Motivating example

Rule component $\Pi$:

\begin{align*}
r_1 &: \text{price-in-usa}(a, \text{high}) \leftarrow \text{made-by}(a, b), \text{NoFellowCompany}(b). \\
r_2 &: \text{price-in-usa}(X, \text{high}) \leftarrow \text{made-by}(X, Y), \text{AmericanAssociate}(Y), \text{monopoly-in-usa}(Y, X). \\
r_3 &: \text{made-by}(a, b). \\
r_4 &: \text{monopoly-in-usa}(b, a).
\end{align*}

DL component $\Sigma$:

\begin{align*}
T-\Box: \\
\text{European} \cap \text{American} \subseteq \bot \\
\text{NoFellowCompany} \equiv \forall \text{associate.} \neg \text{American} \\
\text{EuropeanAssociate} \equiv \exists \text{associate.} \text{American} \\
\text{AmericanAssociate} \equiv \exists \text{associate.} \text{American} \\
\text{InternationalCompany} \equiv \text{EuropeanAssociate} \cup \text{AmericanAssociate} \\
\end{align*}

\begin{align*}
A-\Box: \\
\text{InternationalCompany}(b) \\
\end{align*}

$\Pi \cup \Sigma \models \text{price-in-usa}(a, \text{high})$?
Motivating example

Rule component $\Pi$:

- $r_1$: price-in-usa(X,high) :-
  made-by(X,Y),
  NoFellowCompany(Y).
- $r_2$: price-in-usa(a,high) :-
  made-by(a,b),
  AmericanAssociate(b),
  monopoly-in-usa(b,a).
- $r_3$: made-by(a,b).
- $r_4$: monopoly-in-usa(b,a).

DL component $\Sigma$:

$\Sigma \nRightarrow \neg \text{AmericanAssociate}(b)$

$\Pi \cup \Sigma \models \text{price-in-usa}(a, \text{high})$?
Motivating example

Rule component $\Pi$:

$r_1$: price-in-usa(a, high) :-
    made-by(a, b),
    NoFellowCompany(b).

$r_2$: price-in-usa(a, high) :-
    made-by(a, b),
    AmericanAssociate(b),
    monopoly-in-usa(b, a).

$r_3$: made-by(a, b).
$r_4$: monopoly-in-usa(b, a).

DL component $\Sigma$:

$\sum \models$ NoFellowCompany(b) \lor
AmericanAssociate(b)

$\Pi \cup \Sigma \models$ price-in-usa(a, high) ?
Motivating example

Rule component $\Pi$:

$r_1$: price-in-usa(X, high) :- made-by(X,Y), NoFellowCompany(Y).

$r_2$: price-in-usa(X, high) :- made-by(X,Y), AmericanAssociate(Y), monopoly-in-usa(Y,X).

$r_3$: made-by(a,b).

$r_4$: monopoly-in-usa(b, a).

DL component $\Sigma$:

$T$-Box:

European $\cap$ American $\subseteq \bot$
NoFellowCompany $\equiv \forall$associate.¬American
EuropeanAssociate $\equiv \exists$associate.American
AmericanAssociate $\equiv \exists$associate.American
InternationalCompany $\equiv$ EuropeanAssociate $\cup$ AmericanAssociate

$A$-Box:

InternationalCompany(b)

Thus, $\Pi \cup \Sigma \models $ price-in-usa(a, high)!
Rules we consider

**HEAD** ← **BODY**

- **HEAD** is some basic construct (*atom*)
- **BODY** is a set of atoms
- **Safety**: head variables appear in the body
- Examples:
  - Datalog: atomic formulae
  - Xcerpt: *Query terms* and *Construct terms*
Semantics of rules

- **Fixpoint semantics**
  - Rules derive ground atoms from given ground atoms
  - **matching** of body atoms vs. given atoms gives substitution $\theta$
  - $\theta$ applied to head $\Rightarrow$ derived atom

$$T_p(S) = \{ H\theta \mid (H \leftarrow B_1, \ldots, B_n) \in P \text{ and } \ (B_1, \ldots, B_n) \text{ matches some } A_1, \ldots, A_n \text{ in } S \text{ with result } \theta \}$$

- $T_p$ monotonic, $T_p(S) \subseteq T_p(S')$ for any $S \subseteq S'$
- **Semantics of program $P$**: least fixpoint of $T_p$
Examples of rules languages

The class includes

- Logical rule languages, e.g.
  - **Datalog** (without negation)
  - Semantics of program: set of Datalog atoms
  - Least Herbrand model
- Rule languages lacking logical semantics, e.g.
  - **Xcerpt** (negation-free subset)
  - Semantics of program: set of Xcerpt data terms
Extended rules

\[ \text{HEAD} \leftarrow \text{BODY}, \text{C} \]

- \text{C} formula of an \textit{external theory} in logical language \textit{L}
- Ground atoms associated with a constraint
  - \textit{A;C} where \textit{A} is a ground atom, \textit{C} formula of \textit{L}
- Extend \text{TP} operator:

\[
\text{TP}(S) = \{ \text{H}\theta; (\text{C}\theta\land C_1\land\ldots\land C_n) \mid (\text{H} \leftarrow B_1, \ldots, B_n,C) \in P \text{ and for some } A_1;C_1, \ldots, A_n;C_n \text{ in } S \text{ (} B_1, \ldots, B_n \text{ matches } A_1, \ldots, A_n \text{ with result } \theta \} \]

Semantics of extended rules

- Restrict model of underlying rule program
  - A constraint \( C \), wrt. an external theory \( \Sigma \), can be:

  1. True in all models of \( \Sigma \) (\( \Sigma \models C \))
  2. False in all models of \( \Sigma \) (\( \Sigma \models \neg C \))
  3. None of above: satisfiable, but false in some models of \( \Sigma \)

\[
M(P) = \{ A \mid A \in \text{lfp}(T_P) \text{ and } \Sigma \models C_A \}
\]

- \( C_A \) is a disjunction of all constraints of \( A \)
Example instances

- Existing rule reasoners not aware of “external” predicates
  - How to re-use rule reasoners to collect constraints?
  - Solved specifically for each language and rule reasoner

- Here:

  (1) **Datalog** + OWL DL
  (2) **Xcerpt** + OWL DL
(1) Collecting constraints in XSB for Datalog

\[ \Pi \]

\[ r_1: \text{price-in-usa}(X, \text{high}) \leftarrow \text{made-by}(X, Y), \text{NoFellowCompany}(Y). \]

\[ r_2: \text{price-in-usa}(X, \text{high}) \leftarrow \text{made-by}(X, Y), \text{AmericanAssociate}(Y), \text{monopoly-in-usa}(Y, X). \]

\[ r_3: \text{made-by}(a, b). \]

\[ r_4: \text{monopoly-in-usa}(b, a). \]

\[ \Pi' \]

\[ r_1: \text{price-in-usa}(X, \text{high}, [\text{NoFellowCompany}(Y) \mid A]) \leftarrow \text{made-by}(X, Y, A). \]

\[ r_2: \text{price-in-usa}(X, \text{high}, [\text{AmericanAssociate}(Y) \mid A]) \leftarrow \text{made-by}(X, Y, A_1), \text{monopoly-in-usa}(Y, X, A_2), \text{append}(A_1, A_2, A). \]

\[ r_3: \text{made-by}(a, b, []). \]

\[ r_4: \text{monopoly-in-usa}(b, a, []). \]
(1) Collecting constraints in XSB for Datalog

- **Query** ← \textit{price-in-usa}(a, high, C) wrt. \( \Pi' \):

\[ C = \text{[NoFellowCompany(b)]} \]
\[ C = \text{[AmericanAssociate(b)]} \]

**\( \Pi' \)**

\[ r_1: \text{price-in-usa}(X, \text{high}, \text{[NoFellowCompany(Y) | A]}) \leftarrow \text{made-by}(X, Y, A). \]

\[ r_2: \text{price-in-usa}(X, \text{high}, \text{[AmericanAssociate(Y) | A]}) \leftarrow \text{made-by}(X, Y, A_1), \text{monopoly-in-usa}(Y, X, A_2), \text{append}(A_1, A_2, A). \]

\[ r_3: \text{made-by}(a, b, []). \]
\[ r_4: \text{monopoly-in-usa}(b, a, []). \]

**\( \text{ground}(\Pi) \)**

\[ r_1: \text{price-in-usa}(a, \text{high}) \leftarrow \text{made-by}(a, b), \text{NoFellowCompany(b)}. \]

\[ r_2: \text{price-in-usa}(a, \text{high}) \leftarrow \text{made-by}(a, b), \text{AmericanAssociate(b)}, \text{monopoly-in-usa}(b, a). \]

\[ r_3: \text{made-by}(a, b). \]
\[ r_4: \text{monopoly-in-usa}(b, a). \]
(2) Collecting constraints in Xcerpt

\[ \Pi \]

CONSTRUCT
\[ \text{results } \{ \text{all madeby } \}
\]
\\text{FILTER}
\[ \text{NoFellowCompany } \{ \text{var Manufact} \} \]
\text{FROM}
\[ \text{madeby } \{ \]
\\text{product } \{ \text{var Product} \},
\\text{manufacturer } \{ \text{var Manufact} \}
\]
\text{END}

GOAL
\[ \text{prices } [ \text{all high } [ \text{var Product} ] ] \]
\text{FROM}
\[ \text{results } \{ \]
\\text{or } \{ 
\text{madeby } [ \text{var Product}, \text{var M } ],
\text{monopoly } [ \text{var M, var Product} ] 
\} 
\}
\text{END}

CONSTRUCT
\[ \text{results } \{ \text{all monopoly } \}
\]
\text{FILTER}
\[ \text{AmericanAssociate } \{ \text{var Manufact} \} \]
\text{FROM}
\[ \text{and } \{ 
\text{monopoly } \{
\text{name } [ \text{var Manufact} ],
\text{product } [ \text{var Product} ]
\},
\text{madeby } \{
\text{product } [ \text{var Product} ],
\text{manufacturer } [ \text{var Manufact} ]
\}
\}
\text{END}

\[ \Pi \text{ made-by [ product [ "A" ], manufacturer [ "B" ] ] XML data monopoly [ name [ "B" ], product [ "A" ] ]} \]
(2) Collecting constraints in Xcerpt

CONSTRUCT
results { all madeby {
  var Product, var Manufact
}

FILTER
NoFellowCompany { var Manufact }

FROM
madeby {
  product { var Product },
  manufacturer { var Manufact }
}

END
GOAL
prices [ all high [ var Product ] ]
FROM
results {{
or {
madeby [ var Product, var C1 ], monopoly [ var C2, var Product ]
}}
END

(2) Collecting constraints in Xcerpt

CONSTRUCT
results { all madeby {
var Product, var Manufact
} }
FILTER
NoFellowCompany { var Manufact }
FROM
madeby {
product { var Product },
manufacturer { var Manufact }
}
END

CONSTRUCT
results { all monopoly {
var Manufact, var Product
} }
FILTER
AmericanAssociate { var Manufact }
FROM
and {
monopoly {
name { var Manufact },
product { var Product }
},
madeby {
product { var Product },
manufacturer { var Manufact }
}
}
END

\[ \Pi' \]

\( \Pi \)

made-by [ product [ "A" ], manufacturer [ "B" ] ]

XML data

monopoly [ name [ "B" ], product [ "A" ] ]
(2) Collecting constraints in Xcerpt

```
CONSTRUCT
results { all monopoly { var Manufact, var Product }
}
FILTER
AmericanAssociate { var Manufact }
FROM
and {
  monopoly { name { var Manufact },
    product { var Product }
  },
  madeby { product { var Product },
    manufacturer { var Manufact }
  }
}
END
```

```
GOAL
prices [ all high [ var Product ] ]
FROM
results {{
  or {
    madeby [ var Product, var M ],
    monopoly [ var M, var Product ]
  }
}}
END
```

```
GOAL
prices [ all high [ var Product, digor [ all var C ] ] ]
FROM
results {{
  or {
    madeby [ var Product, var M, var C ],
    monopoly [ var M, var Product, var C ]
  }
}}
END
```

```
made-by [ product [ "A" ], manufacturer [ "B" ] ]
XHTML data
monopoly [ name [ "B" ], product [ "A" ] ]
```
(2) Collecting constraints in Xcerpt

Query wrt. \( \Pi' \):

\[
\Sigma \models \text{NoFellowCompany}(B) \lor \text{AmericanAssociate}(B)
\]
Implementing integrated reasoner with Xcerpt

1. Compile **Extended Xcerpt** (\( \Pi \)) programs into plain Xcerpt (\( \Pi' \))

\[
\Pi \rightarrow \Pi'
\]

- Collect all constraints related to an Xcerpt answer using the **all** construct

2. Run \( \Pi' \) in existing Xcerpt engine, returning answers and (boolean) DL queries (in DIG syntax)

3. Submit DL queries to a DL reasoner

4. Return Xcerpt answers for which the DIG query is “true”
Eager interaction

**Rule component \( \Pi \):**

\begin{align*}
r_1: & \text{price-in-usa}(X, \text{high}) :- \text{made-by}(X, Y), \text{American}(Y), \text{monopoly-in-usa}(Y, X). \\
r_2: & \text{price-in-usa}(X, \text{high}) :- \text{made-by}(X, Y), \text{European}(Y). \\
r_3: & \text{made-by}(a, b). \\
r_4: & \text{monopoly-in-usa}(b, a).
\end{align*}

**DL component \( \Sigma \):**

\begin{align*}
\text{T-Box:} & \quad \text{European} \cap \text{American} \subseteq \bot \\
\text{A-Box:} & \quad \text{...} \\
\text{European(b)} & \end{align*}
Eager interaction

Q: \text{price-in-usa}(a, \text{high})

\begin{align*}
\text{made-by}(a, b), \\
\text{American}(b), \\
\text{monopoly-in-usa}(b, a)
\end{align*}

\begin{align*}
\text{made-by}(a, b), \\
\text{European}(b)
\end{align*}

\begin{align*}
\text{American}(b), \\
\text{monopoly-in-usa}(b, a)
\end{align*}

\begin{align*}
\text{European}(b)
\end{align*}

\[ \Sigma \models \neg \text{American}(b) \]  
\[ \Rightarrow \text{Prune} \]

Rule component \( \Pi \):

\begin{align*}
\text{r}_1: & \text{price-in-usa}(X, \text{high}) \leftarrow \\
& \text{made-by}(X, Y), \\
& \text{American}(Y), \\
& \text{monopoly-in-usa}(Y, X).
\end{align*}

\begin{align*}
\text{r}_2: & \text{price-in-usa}(X, \text{high}) \leftarrow \\
& \text{made-by}(X, Y), \\
& \text{European}(Y).
\end{align*}

\begin{align*}
\text{r}_3: & \text{made-by}(a, b).
\end{align*}

\begin{align*}
\text{r}_4: & \text{monopoly-in-usa}(b, a).
\end{align*}

DL component \( \Sigma \):

\begin{align*}
T-\text{Box}: \\
\text{European} \cap \text{American} & \subseteq \bot
\end{align*}

\begin{align*}
A-\text{Box}: & \ldots \\
\text{European}(b)
\end{align*}
Eager interaction

Q: 

price-in-usa(a, high)

made-by(a, b),
European(b)

European(b)

Rule component $\Pi$:

\begin{align*}
  r_1: & \text{price-in-usa}(X, \text{high}) :\iff \\
        & \text{made-by}(X, Y), \\
        & \text{American}(Y), \\
        & \text{monopoly-in-usa}(Y, X). \\
  r_2: & \text{price-in-usa}(X, \text{high}) :\iff \\
        & \text{made-by}(X, Y), \\
        & \text{European}(Y). \\
  r_3: & \text{made-by}(a, b). \\
  r_4: & \text{monopoly-in-usa}(b, a).
\end{align*}

DL component $\Sigma$:

\begin{align*}
  T-\Box: & \text{European} \cap \text{American} \subseteq \bot \\
  A-\Box: & \ldots \\
  \text{European}(b)
\end{align*}
Prototype: Datalog + OWL DL

• Interfaces existing reasoners
  – Rule reasoner: XSB
  – Ontology reasoner: DIG compliant DL reasoner
    • Only OWL concepts
    • Possible extension:
      – Allow roles in constraints through “rolling-up”
      – Eager interaction
        • No pruning yet...

– Available at: http://www.ida.liu.se/hswrl
Conclusions

- Combining general class of rules with constraints
  - Rules are negation-free, fixpoint semantics
- Non-logical rule-languages
  - E.g. Xcerpt
- Re-using existing reasoners
- Prototype integration
  - Datalog + OWL-DL
  - Using XSB + RacerPro
Our work

- is motivated by and extends AL-Log \cite{Donini et. al.}
- aims at integration of existing reasoners
  - not restricted to Datalog
- supports reasoning by cases in DL, unlike:
  - ASP+DL \cite{Eiter et. al.}
    - Handles negation, supports bi-directional flow of information between rules and DL KBs
- does not extend ontology languages like, e.g.:
  - SWRL \cite{Horrocks et. al.}, OWL-DL \cite{Motik et. al.}, Safe Hybrid KBs \cite{Rosati}
Future work

- How to re-use existing rule reasoners?
- Eager interaction
  - Practical use-cases and full implementation
- Other constraint languages
- Rules with negation
  - Well-founded semantics
The End

Thank you!